L3: Entry 6 of 29 File: USPT Oct 21, 2003

DOCUMENT-IDENTIFIER: US 6636922 B1
TITLE: Methods and apparatus for implementing a host side advanced serial protocol

Brief Summary Text (13):
In yet a further embodiment, a system for executing data transfers between a host computer and a peripheral device over a serial link is disclosed. The system includes a CPU for setting up a command descriptor block (CDB) with an ATA/ATAPI command and a scatter/gather list in a system memory. A pointer register that is resident in a host controller chip is also provided, and the CPU is configured to write an address of the CDB into the pointer register. A DMA controller is provided for instructing a transfer of the CDB to RAM memory of the host controller chip, and the host controller chip is configured to send the CDB to a peripheral device over a serial physical cable. The host controller chip further being configured to send IN packets to poll the peripheral device for data packets. The device is configured to respond to the poll by sending data packets to the RAM of the host controller chip. A data buffer is also provided and is configured to receive the data packets at the direction of the DMA controller, and the data buffer being pointed to the scatter/gather list in the system memory. The host controller chip is configured to set up an interrupt status to interrupt the CPU indicative of the receipt of the data packets from the peripheral device.

Detailed Description Text (30):
FIG. 8 shows a more detailed diagram of the ASP Host Controller 104a, which consists of a system bus interface and an ASP bus interface. These two interfaces provide data flow between the host and the ASP devices 106. In one preferred embodiment, the ASP Host Controller 104a can support one outstanding command per device. The system bus interface allows the system software by way of a system bus 164 to communicate with the ASP devices. It provides a mechanism to transfer data to and from the host memory, and a mechanism to send notifications to the system software. In one embodiment, it uses a DMA controller to transfer data to and from the host memory, and uses an interrupt to send a notification to the system software at successful command completion or abort due to errors. In one preferred embodiment, the Host Controller also supports scatter/gather data transfers. The ASP Host Controller 104a supports a mechanism to accept ATA/ATAPI commands. The system bus interface may contain some configurable parameters. Some of these configurable parameters include: (a) capabilities of the host controller; (b) configurable link speeds, packet sizes, and repeat counts per device; (c) globally and individually maskable interrupts, and interrupt status; (d) device attachment information; and (e) mechanisms to start a command execution and data transfer.

Detailed Description Text (34):
In accordance with the present invention, ASP can issue one outstanding command per device to the ASP Host Controller 104a. Thus, all devices can overlap their command executions. The HCD also provides the memory addresses and transfer counts to the ASP Host Controller 106, and the data transfer is performed by the DMA controller. Also, the value at offset 0, which was used for PIO data transfers, is no longer used. The use of DMA commands and the use of LBA addressing instead of CHS addressing are now used. The following data elements contain the relevant information for a request: (a)an ATA command and optionally an ATAPI command, and an ATA status; (b) data transfer information indicating the addressed device,

whether data transfer is involved, and the direction of the data transfer; and (c) physical memory addresses and transfer counts of a list of scatter/gather data buffers.

Detailed Description Text (205):
FIG. 42 shows a command execution flow400 of the present invention which involves the following operations: 1. The CPU 402 sets up a Command Descriptor Block (CDB) 404 with an ATA/ATAPI command and a scatter/gather list in system memory. 2. The CPU 402 writes the address of this CDB to the CDB Pointer Register of the ASP HCI 408. 3. The host controller sets the DMA controller to transfer the CDB to internal memory. 4. The host controller 408 sends the command to the device and sends IN packets to poll the device 106 for data. 5. The device responds with data packets. 6. The host controller sets up the DMA controller to transfer the data to the data buffers 406 pointed to by the scatter/gather list. 7. The host controller 408 sets up the interrupt status and interrupts the CPU 402.

Detailed Description Text (207):
The HCI data structures include a command descriptor, a data transfer descriptor, and a scatter/gather list. The command descriptor consists of an ATA and ATAPI command. The data transfer descriptor specifies the device address, and data transfer direction and count. The scatter/gather list specifies a list of physical memory addresses and transfer counts for data transfers. In a preferred embodiment, the Host Controller data structure should be in one contiguous block of double-word aligned physical memory. The Host Controller is given a pointer to this data structure to start a command execution. The following Tables 44 and 45 illustrate a format of the Host Command Descriptor Block (CDB).

Detailed Description Text (216):
The scatter/gather list specifies the physical memory addresses and sizes of the data buffers in system memory, and may contain any number of entries. The EOT bit indicates the end of this list. However, a maximum of 16 segment entries is recommended. Physical memory addresses can be byte-aligned. The maximum transfer count is 64K, and the count should be an even number, and zero indicates 64K. EOT=End of table.

Detailed Description Paragraph Table (44):
TABLE 44 Host Command Descriptor Block for 32-bit Addressing Offset Length Description Type 0 8 ATA Command/ATA Status Command 8 16 ATAPI Command Descriptor 24 4 Data Control Data Transfer 28 4 Total Transfer Count Descriptor 32 4 Segment 0 Physical Memory Address Scatter/Gather 36 4 Segment 0 Transfer Count List 40 4 Segment 1 Physical Memory Address 44 4 Segment 1 Transfer Count . . .

Detailed Description Paragraph Table (45):
TABLE 45 Host Command Descriptor Block for 32-bit Addressing Offset Length Description Type 0 8 ATA Command/ATA Status Command 8 16 ATAPI Command Descriptor 24 4 Data Control Data Transfer 28 4 Total Transfer Count Descriptor 32 8 Segment 0 Physical Memory Address Scatter/Gather 40 4 Segment 0 Transfer Count List 44 8 Segment 1 Physical Memory Address 52 4 Segment 1 Transfer Count . . .

Detailed Description Paragraph Table (52):
TABLE 52 Scatter/Gather Entry for 32-bit Addressing (8 bytes) Byte 3 Byte 2 Byte 1 Byte 0 Physical Memory Address EOT Reserved Transfer Count [15:1] 0

Detailed Description Paragraph Table (53):
TABLE 53 Scatter/Gather Entry for 64-bit Addressing (12 bytes) Byte 3 Byte 2 Byte 1 Byte 0 Physical Memory Address [31:00] Physical Memory Address [63:32] EOT Reserved Transfer Count [15:1] 0

CLAIMS:

23. A system for executing data transfers between a host computer and a peripheral device over a serial link, comprising: a CPU for setting up a command descriptor block (CDB) with an ATA/ATAPI command and a scatter/gather list in a system memory; a pointer register resident in a host controller chip, the CPU being configured to write an address of the CDB into the pointer register; a DMA controller for instructing a transfer of the CDB to RAM memory of the host controller chip; the host controller chip being configured to send the CDB to a peripheral device over a serial physical cable, the host controller chip further being configured to poll the peripheral device for data packets; the device being configured to respond to the poll by sending data packets to the RAM of the host controller chip; a data buffer being configured to receive the data packets at the direction of the DMA controller, the data buffer being pointed to the scatter/gather list in the system memory; and the host controller chip being configured to set up an interrupt status to interrupt the CPU indicative of the receipt of the data packets from the peripheral device.